FIG. 1

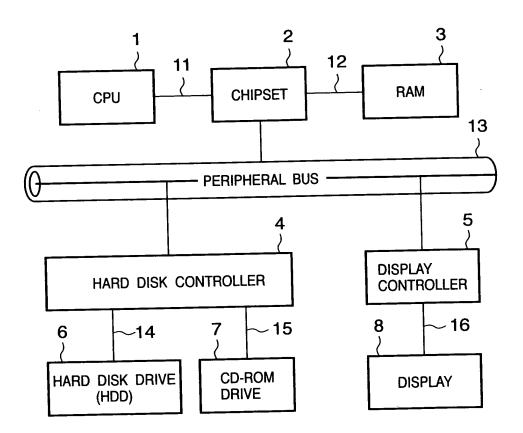


FIG. 2

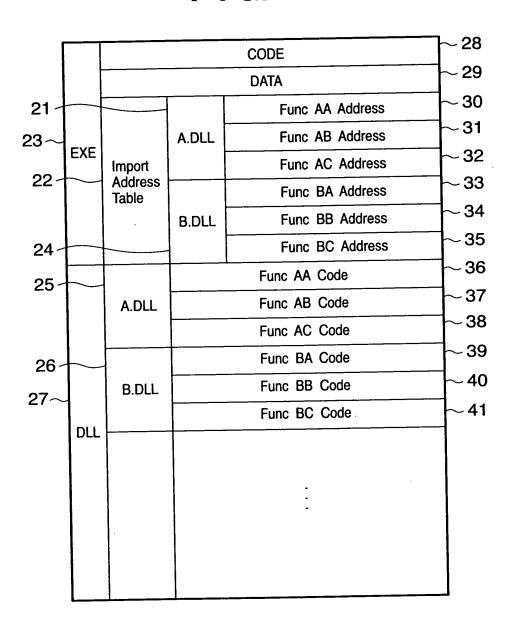
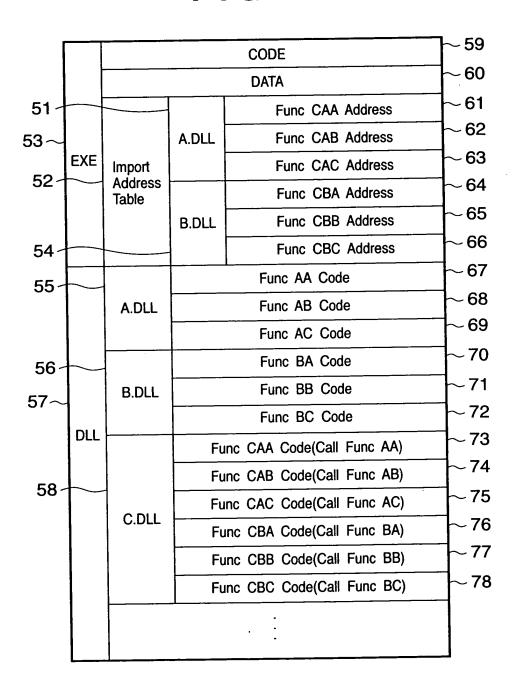


FIG. 3



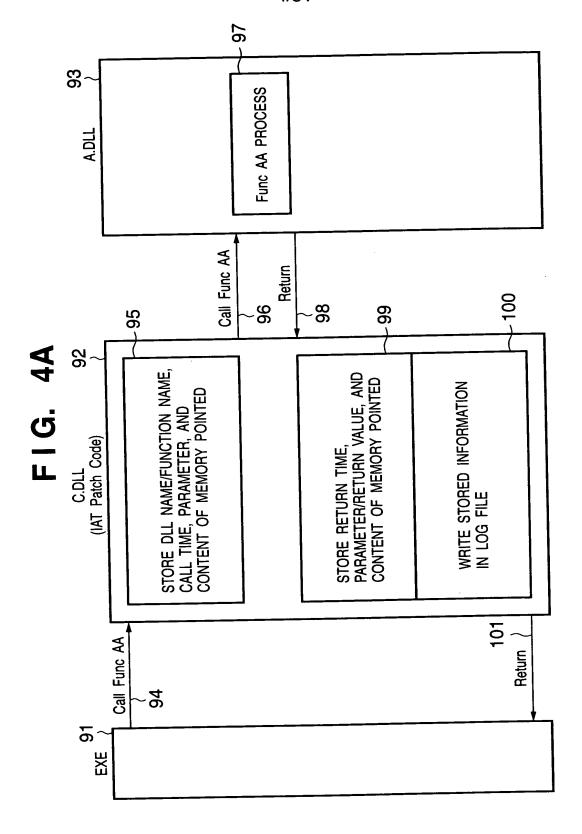


FIG. 4B

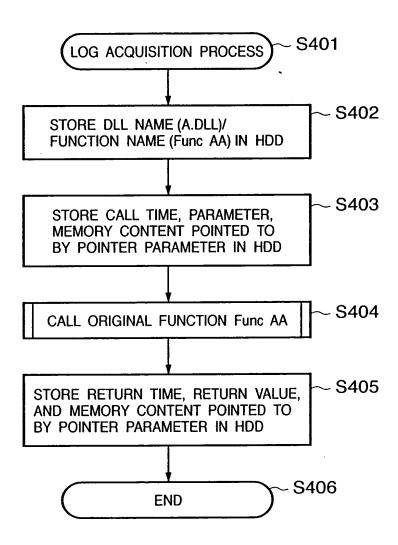


FIG. 5

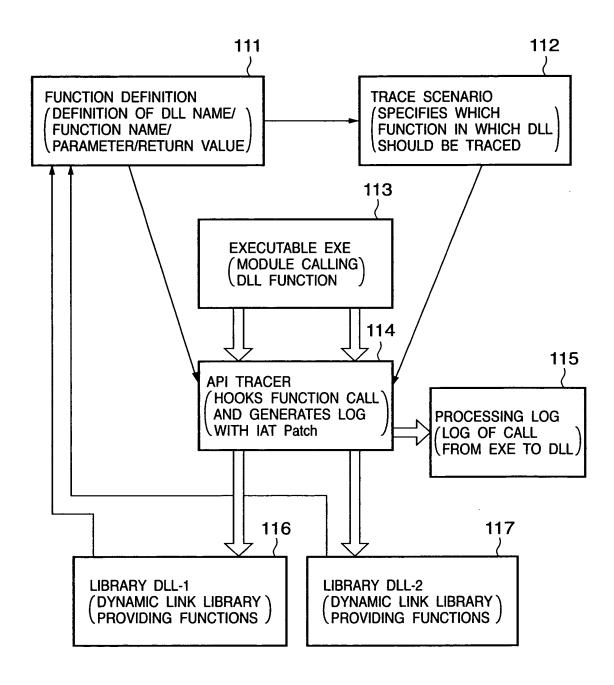
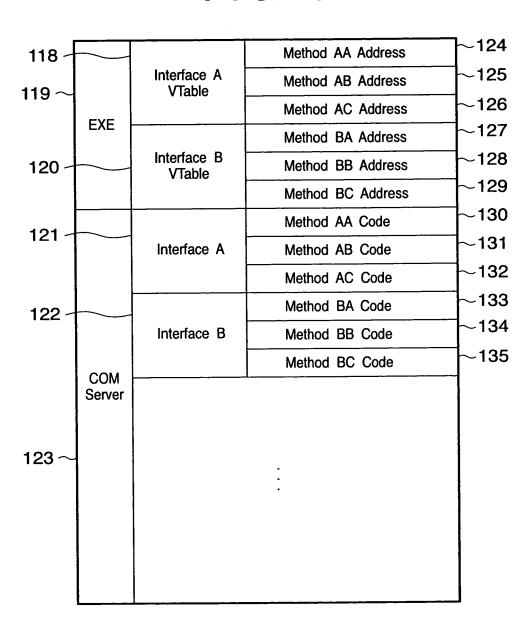


FIG. 6



| 136 | | | | | |
|--|--------------|------|--------------|---|------------------------------|
| Nethod A'C Address | 136 | | | Method A'A Address | ~145 |
| SEXE | 107 - | | | Method A'B Address | ─ ~146 |
| Interface B' Method B'A Address 148 Method B'B Address 150 Method AA Code 151 Interface A Method AB Code 152 Method AC Code 153 Method BC Code 154 Interface B Method BB Code 155 Method BC Code 155 Method BC Code 156 Method BC Code 156 Interface A' Method A'A Code (Call Method AB) 158 Method A'C Code (Call Method AB) 159 Method B'A Code (Call Method AC) 160 Method B'B Code (Call Method BA) 160 Method B'B Code (Call Method BA) 160 Method B'B Code (Call Method BB) 160 Method B'B Code (Call Method BB) 160 Method B'B Code (Call Method BB) 161 | 137~ | | | Method A'C Address | ~147 |
| 138 | | EXE | VTable _ | Method B'A Address | ~148 |
| Method B'C Address | 138 | | | Method B'B Address | ~149 |
| Interface A Method AB Code COM Server Interface B Method BB Code Co | | : | | Method B'C Address | ~150 |
| Interface A Method AB Code — 152 Method AC Code — 153 Method BA Code — 154 Interface B Method BB Code — 155 Method BC Code — 156 Method BC Code — 157 Method A'A Code (Call Method AA) Interface A' Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'B Code (Call Method BB) Method B'B Code (Call Method BB) Method B'C Code — 162 | 100 | | | Method AA Code | ~151 |
| COM Server Interface B Method BA Code 154 141 Method BB Code 155 140 Method BC Code 156 Method BC Code 156 Method A'A Code (Call Method AA) 157 Method A'B Code (Call Method AB) 158 Method A'C Code (Call Method AC) 159 Method B'A Code (Call Method BA) 160 Method B'B Code (Call Method BA) 161 Method B'B Code (Call Method BB) 161 Method B'C Code 162 | 139 | | | Method AB Code | ~152 |
| Interface B Method BB Code 155 Method BC Code 156 140 Method BC Code 157 Method BC Code 157 Method A'A Code (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) DLL Method B'A Code (Call Method BA) Method B'A Code (Call Method BA) Method B'B Code (Call Method BA) Method B'B Code (Call Method BB) Method B'B Code (Call Method BB) Method B'C Code 157 158 Method B'C Code 160 161 | | | | Method AC Code | ~153 |
| Interface B Method BB Code Method BC Code Method BC Code Method A'A Code (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Method B'A Code (Call Method BA) Method B'B Code (Call Method BA) Method B'B Code (Call Method BB) Method B'C Code 161 | | | Interface B | Method BA Code | ~154 |
| 140 - : : : : : : : : : : : : : : : : : : | | | | Method BB Code | ~155 |
| Method A'A Code (Call Method AA) Interface A' Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'C Code 161 Method B'C Code | 141 - | | | Method BC Code | [−] ~156 |
| Method A'A Code (Call Method AA) Interface A' Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'C Code 161 Method B'C Code | | | L | | |
| Interface A' Method A'B Code (Call Method AB) | | | | • | 1 |
| Interface A' Method A'B Code (Call Method AB) | 140 <i>~</i> | | | : | |
| Interface A' (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'C Code 162 | 140~ | | | | ~157 |
| Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'C Code 162 | | | | | ~157 |
| Call Method AC) Method B'A Code (Call Method BA) Method B'B Code (Call Method BB) Method B'C Code 162 | | | Interface A' | (Call Method AA) Method A'B Code | |
| Method B'A Code (Call Method BA) Interface B' Method B'B Code (Call Method BB) Method B'C Code ~162 | | | Interface A' | (Call Method AA) Method A'B Code | |
| Method B'A Code (Call Method BA) Method B'B Code (Call Method BB) Method B'C Code ~162 | | | Interface A' | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code | ~158 |
| Interface B' Method B'B Code (Call Method BB) Method B'C Code ~162 | | DLL | Interface A' | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code | ~158 ~159 |
| Interface B' (Call Method BB) Method B'C Code ~162 | 142 | DLL | Interface A' | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code | ~158 ~159 |
| Method B'C Code ~162 | 142 | DLL | Interface A' | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code | ~158 ~159 |
| | 142 | DLL. | | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Method B'B Code | ~158 ~159 ~160 |
| (04.1.1.04.1.04.1.04.1.04.1.04.1.04.1.04 | 142 | DLL | | (Call Method AA) Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Method B'B Code | ~158 ~159 ~160 |
| | 142 | DLL | | Method A'B Code (Call Method AB) Method A'C Code (Call Method AC) Method B'A Code (Call Method BA) Method B'B Code (Call Method BB) Method B'C Code | ~158 ~159 ~160 ~161 |

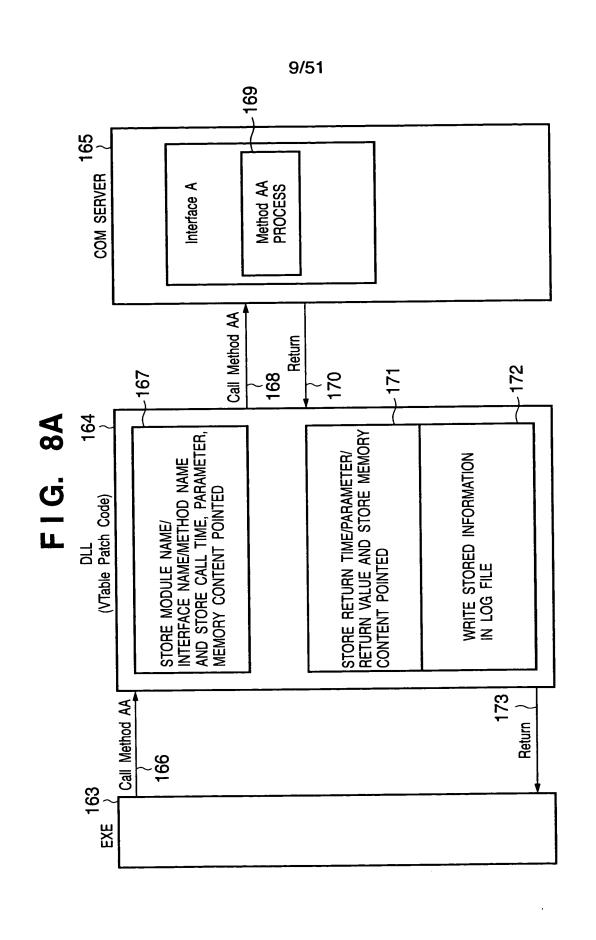


FIG. 8B

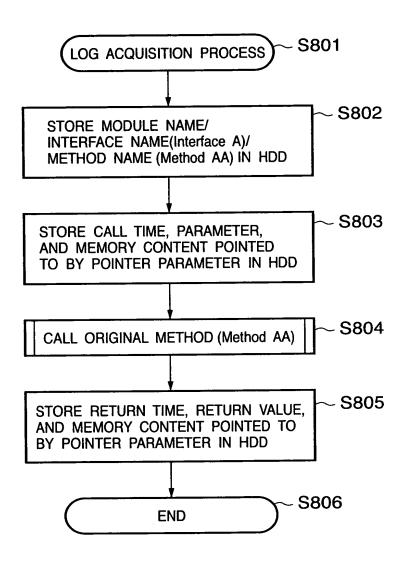
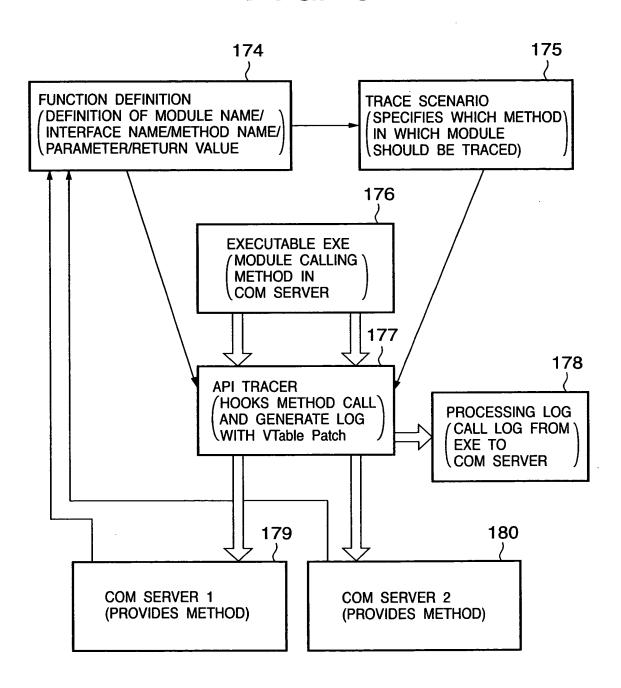


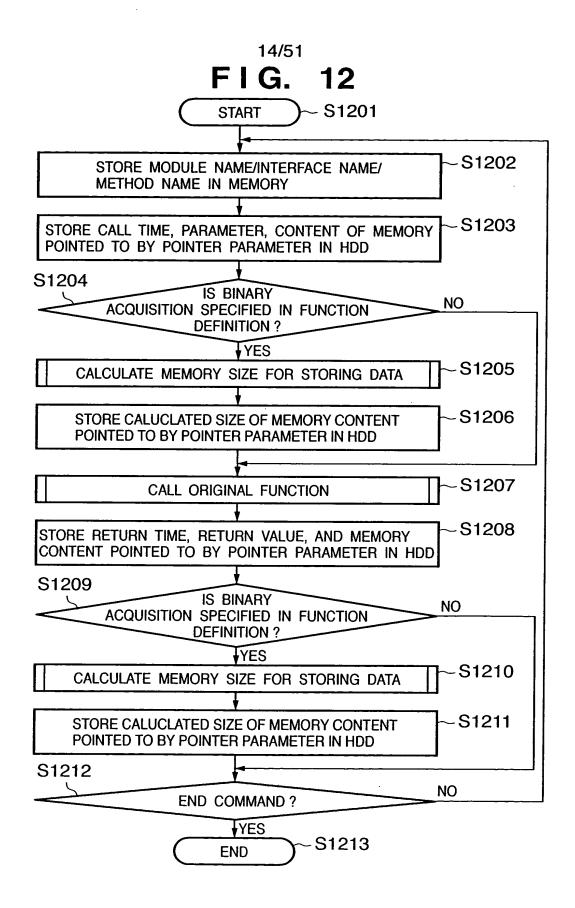
FIG. 9



. .

```
//TestDIIStd
     uuid(58DB5633-0694-4340-97CE-4E1AC6BFFBA7),
     helpstring("TestDIIStd Type Library For PAT"),
     version(1,0)
library TestDIIStd
        typedef [public] struct
                 char chParam;
                 unsigned char uchParam;
                 short sParam;
                 unsigned short usParam;
                 int nParam;
                 unsigned int unParam;
                 long IParam;
                 unsigned long ulParam;
                 double dbParam;
                 float fParam;
         }TESTSTRUCT;
        typedef [public] TESTSTRUCT *LPTESTSTRUCT;
//DEFINE_GUID(GUID_PROGID, 0x8e037d65, 0xefa0, 0x40e7, 0x91, 0x43, 0xef, 0x70, 0x56,
0x94, 0x5b, 0x79);
      uuid(8E037D65-EFA0-40e7-9143-EF7056945B79),
      helpstring("TestDIIStd.dll for PAT object"),
         interface
         test
         {
                 char stdcall FuncCharStd([in] char chPram);
                 char* stdcall FuncPCharStd([in, out] char* lpchParam);
                 TESTSTRUCT_stdcall FuncStructStd[in]TESTSTRUCT TestStruct);
                  LPTESTSTRUCT\_stdcall \ \ FuncPStructStd([in, \ out]LPTESTSTRUCTIp \ \ TestStruct): \\
        }:
```

```
200
interface
test
{
                                                        201
       void_stdcall FuncBinidls
               [out, custom(PAT_PARAM_ATTR_ID, "binid_is()")] long* lpIParam
       void_stdcall FuncSizels
                                                         202
               [in] DWORD dwCount,
               [out, custom(PAT_PARAM_ATTR_ID, "sizeis_is(dwCount)")] int* lpnParam
       void_stdcall FuncLengthls
                                                         203
               [in] DWORD dwLength,
               [in, custom(PAT_PARAM_ATTR_ID, "lentgth_is(dwLength)")] char* lpszParam
       void_stdcall FuncBytesIs
                                                         204
               [in] DWORD dwSize,
              [in, custom(PAT_PARAM_ATTR_ID, "bytes_is(dwSize)")] void* IpParam
                                                     205
       void_stdcall FuncBytesls2
              [out, custom(PAT_PARAM_ATTR_ID, "bytes_is(12)")] void* IpParam
       );
};
```



| | | 15/51 | | |
|---|--|---|--|------|
| DataID: 0x0001 Size: 4 00000000: 10 00 00 DataID: 0x0002 Size: 40 | 00000000 : 05 00 00 00 4A 03 A5 20 00000008 : 06 00 00 00 4B 03 A5 20 00000010 : 07 00 00 00 4C 03 A5 20 00000018 : 08 00 00 00 4E 03 A5 20 00000020 : 09 00 00 00 4E 03 A5 20 DatalD : 0x0003 | Size:5 00000000:66 4A 70 50 00 DatalD:0x0004 00000000:015D 66 B2 20 49 20 | 211 | ~210 |
| TestDIIStd. DLL FuncBinidIs long* IpParam: 0x5034206D/0x10, DataID=0x0001 void: 2002/03/25 22: 24: 12.025 2002/03/25 22: 24: 12.035 | TestDIIStd. DLL FuncSizels DWORD dwCount: 10 int* lpnParam: 0x5034207D/0x5, DataID=0x0002 void: 2002/03/25 22: 24: 12.046 2002/03/25 22: 24: 12.057 | TestDIIStd. DLL FuncLengthls DWORD dwLength:5 char* lpszParam:0x503860C/0x66, DataID=0x0003 void: 2002/03/25 22:24:12.068 2002/03/25 22:24:12.079 | TestDIIStd. DLL FuncBytesIs DWORD dwSize: 7 void* IpParam: 0x503870C/, DataID=0x0004 void: 2002/03/25 22: 24: 12.100 2002/03/25 22: 24: 12.179 | |
| MODULE NAME: FUNCTION NAME: ARGUMENT(in): ARGUMENT(out): ARGUMENT(out): RETURN VALUE: IN TIME: OUT TIME: | MODULE NAME: FUNCTION NAME: ARGUMENT(in): ARGUMENT(out): ARGUMEN VALUE: IN TIME: OUT TIME: | MODULE NAME: FUNCTION NAME: ARGUMENT(in): ARGUMENT(out): RETURN VALUE: IN TIME: OUT TIME: | MODULE NAME: FUNCTION NAME: ARGUMENT(in): ARGUMENT(out): RETURN VALUE: IN TIME: OUT TIME: | : |
| | | Ŕ | | |

```
#define PAT_PARAM_ATTR_ID 00000000-0000-0000-000000000000
typedef [public] struct ~ 220
     [in, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncInternal1)")] DWORD pfnFuncInternal1;
     [in, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncInternal2)")] DWORD pfnFuncInternal2;
     [in, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncInternal3)")] DWORD pfnFuncInternal3;
     [in, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncInternal4)")] DWORD pfnFuncInternal4;
}FUNCPOINTERARRAY;
interface
test
                                                                         221
     void_stdcall SetCallBack
          [in, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncCallBack)")] DWORD
pfnFuncCallBack
     void FuncCallBack([in] int nParam); 222
                                                              223
     void_stdcall GetFuncPointer
          [out, custom(PAT_PARAM_ATTR_ID, "funcname_is(FuncInternal)")] DWORD
pfnFuncInternal
     void FuncInternal([in, out] char* lpszParam); 224
     void_stdcall GetFuncPointerArray
          [out]FUNCPOINTERARRAY* pFuncPointerArray; ~ 225
     void FuncInternal1([in] int nParam);
     void FuncInternal2([in, out] char* lpzaParam);
     void FuncInternal3([out] DWORD* dwParam); ~ 226
     void FuncInternal4([);
```

FIG. 15

| | | | | CODE | 7 | |
|-----|-----|----------------------------|-----------------------------|------------------|------|--|
| 230 | EXE | CODE | | | - | |
| | | | | Func CAA Address | 1 | |
| | | Import Address Table | A.DLL | Func CAB Address | - | |
| | | | | Func CAC Address | 1 | |
| | | | B.DLL | Func CBA Address | 1 | |
| | | | | Func CBB Address | 1 | |
| | | | | Func CBC Address | | |
| | DLL | A.DLL | Func AA Code | | | |
| | | | Func AB Code | | | |
| | | | Func AC Code | | | |
| | | | Func AD Code | | ~231 | |
| | | B.DLL | Func BA Code | | | |
| | | | Func BB Code | | | |
| | | | Func BC Code | | | |
| | | C.DLL. | Func CAA Code(Call Func AA) | | | |
| | | | Func CAB Code(Call Func AB) | | | |
| | | | Func CAC Code(Call Func AC) | | | |
| | | | Func CBA Code(Call Func BA) | | | |
| | | | Func CBB Code(Call Func BB) | | | |
| | | | Func CBC Code(Call Func BC) | | | |
| | | | Fun | ~232 | | |
| | | • | | | | |
| Į | | · | | | | |

18/51 FIG. 16 - S1601 START -S1602 STORE MODULE NAME/INTERFACE NAME/ METHOD NAME IN HDD -S1603 STORE CALL TIME, PARAMETER, AND MEMORY CONTENTS POINTED TO BY POINTER PARAMETER IN HDD S1604 IS funcnmame is NO SPECIFIED IN FUNCTION DEFINITION? **¥YES** S1605 GENERATE LOG ACQUISITION CODE STORE VALUE DEFINED IN funcname_is AND REPLACE S1606 IT WITH ADDRESS OF LOG ACQUISITION CODE -S1607 CALL ORIGINAL FUNCTION -S1608 STORE RETURN TIME, RETURN VALUE, AND MEMORY CONTENTS POINTED TO BY POINTER PARAMETER IN HDD S1609 IS funchame_is NO SPECIFIED IN FUNCTION DEFINITION? YES S1610 GENERATE LOG ACQUISITION PROCESS CODE ~S1611 STORE VALUE DEFINED IN funcname_is AND REPLACE IT WITH POINTER TO LOG ACQUISITION CODE S1612 NO **END COMMAND? ♦YES** -S1613 END

MODULE NAME:

TestDIIStd. DLL

FUNCTION NAME:

SetCallBack

ARGUMENT(in):

DWORD pfnFuncCallBack: 0x0299103F

ARGUMENT(out):

RETURN VALUE:

void:

IN TIME: OUT TIME: 2002/03/25 22:24:12.025 2002/03/25 22:24:12.035

MODULE NAME:

FUNCTION NAME:

TestDIIStd. DLL GetFuncPointer

ARGUMENT(in):

ARGUMENT(out):

DWORD pfnFuncInternal: 0x29913dF

RETURN VALUE:

IN TIME: OUT TIME:

2002/03/25 22:24:12.046 2002/03/25 22:24:12.057

MODULE NAME: FUNCTION NAME:

TestDIIStd. DLL GetFuncPointerArray

ARGUMENT(in):

ARGUMENT(out):

FUNCPOINTERARRAY* pFuncPointerArray: 0x503860C

DWORD FUNCPOINTERARRAY. pfnFuncInternal1:0x02997670 DWORD FUNCPOINTERARRAY. pfnFuncInternal2:0x02997708 DWORD FUNCPOINTERARRAY. pfnFuncInternal3:0x029977BE DWORD FUNCPOINTERARRAY. pfnFuncInternal4:0x0299784F

RETURN VALUE:

void:

IN TIME: OUT TIME:

2002/03/25 22:24:12.068 2002/03/25 22:24:12.079

20/51

FIG. 18

TestDIIStd. DLL MODULE NAME: **FUNCTION NAME:** SetCallBack DWORD pfnFuncCallBack: 0x0299103F ARGUMENT(in): ARGUMENT(out): RETURN VALUÉ: void: 2002/03/25 22:24:12.025 IN TIME: 2002/03/25 22:24:12.035 OUT TIME: MODULE NAME: TestDIIStd. DLL **FUNCTION NAME:** FuncCallBack ARGUMENT(in): int nParam: 100 ARGUMENT(out): RETURN VALUE: void: IN TIME: OUT TIME:

2002/03/25 22:24:12.036 2002/03/25 22:24:12.040

MODULE NAME: TestDIIStd. DLL **FUNCTION NAME: GetFuncPointer**

ARGUMENT(in):

DWORD pfnFuncInternal: 0x029913dF ARGUMENT(out):

RETURN VALUÉ: void:

2002/03/25 22:24:12.046 IN TIME: 2002/03/25 22:24:12.057 OUT TIME:

MODULE NAME: TestDIIStd. DLL **FUNCTION NAME:** FuncInternal1

ARGUMENT(in): char* lpszParam: 0x5038600/0 char* lpszParam: 0x5038600/-12 ARGUMENT(out):

RETURN VALUÉ: void:

IN TIME: 2002/03/25 22:24:12.060 2002/03/25 22:24:12.065 OUT TIME:

MODULE NAME: TestDIIStd. DLL **FUNCTION NAME:** GetFuncPointArray

ARGUMENT(in):

FUNCPOINTERARRAY* pFuncPointerArray: 0x503860C ARGUMENT(out):

DWORD FUNCPOINTERARRAY. pfnFuncInternal1: 0x02997670
DWORD FUNCPOINTERARRAY. pfnFuncInternal2: 0x02997708
DWORD FUNCPOINTERARRAY. pfnFuncInternal3: 0x029977BE DWORD FUNCPOINTERARRAY. pfnFuncInternal4: 0x0299784F

RETURN VALUE:

IN TIME: 2002/03/25 22:24:12.068 OUT TIME: 2002/03/25 22:24:12.079

MODULE NAME: TestDIIStd. DLL FuncInternal4 **FUNCTION NAME:**

ARGUMENT(in): ARGUMENT(out):

RETURN VALUE: void:

IN TIME: 2002/03/25 22 : 24 : 12.080 OUT TIME: 2002/03/25 22:24:12.099

22/51 FIG. **20** - S2001 START S2002 STORE MODULE NAME/INTERFACE NAME/ METHOD NAME IN HDD S2003 IS VARIABLE-LENGTH ARRAY NO ACQUISITION SPECIFIED IN FUNCTION **DEFINITION? ¥YES** -S2004 TREAT POINTER PARAMETER VARIABLE **DEFINITIONS SPECIFIED AS ARRAY** -S2005 STORE CALL TIME, PARAMETER, AND MEMORY CONTENT POINTED TO BY POINTER PARAMETER IN HDD -S2006 CALL ORIGINAL FUNCTION S2007 IS VARIABLE-LENGTH ARRAY NO ACQUISITION SPECIFIED IN FUNCTION **DEFINITION?** YES -S2008 TREAT POINTER PARAMETER VARIABLE DEFINITIONS SPECIFIED AS ARRAY -S2009 STORE RETURN TIME, RETURN VALUE, AND MEMORY CONTENT POINTED TO BY POINTER PARAMETER IN HDD S2010 NO **END COMMAND?** YES S2011 **END**

MODULE NAME: TestDIIStd. DLL FUNCTION NAME: FuncArrayIs DWORD dwCount: 4 ARGUMENT(in): int* IpnParam: 0x5034206D/0x00 ARGUMENT(out) : int* lpnParam: 0x5034206D/0x01 RETURN VALUE: void: IN TIME: 2002/03/25 22:24:12.025 OUT TIME: 2002/03/25 22:24:12.035 250 MODULE NAME: TestDIIStd. DLL **FUNCTION NAME: FuncArrayIs** DWORD dwCount: 3 ARGUMENT(in): int* IpnParam: 0x5034207D/0x00 ARGUMENT(out): int* lpnParam: 0x5034207D/0x05 RETURN VALUE: void: IN TIME: 2002/03/25 22:24:12.046 OUT TIME: 2002/03/25 22:24:12.057 251 MODULE NAME: TestDIIStd. DLL FUNCTION NAME: FuncArrayIs ARGUMENT(in): DWORD dwCount: 4 int* IpnParam: 0x5034206D/Array (int: 0:0x00, int: 1:0x00, int: 2:0x00, int: 3:0x00) ARGUMENT(out) : int* lpnParam: 0x5034206D/Array (int: 0:0x01, int: 1:0x02, int: 2:0x03, int: 3:0x04) RETURN VALUE: void: IN TIME: 2002/03/25 22:24:12.025 OUT TIME: 2002/03/25 22:24:12.035 TestDIIStd. DLL MODULE NAME: FUNCTION NAME: FuncArrayIs ARGUMENT(in): DWORD dwCount: 3 int* lpnParam: 0x5034207D/Array (int: 0:0x00, int: 1:0x00, int: 2:0x00)

ARGUMENT(out): int* lpnParam: 0x5034207D/Array (int: 0: 0x05, int: 1: 0x10, int: 2: 0x15)
RETURN VALUE: void:

2002/03/25 22 : 24 : 12.046 2002/03/25 22 : 24 : 12.057

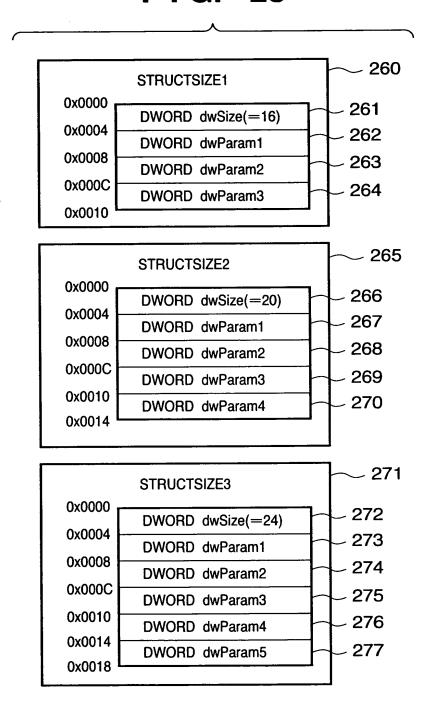
•••

IN TIME:

OUT TIME:

```
typedef struct
       DWORD dwSize;
       DWORD dwParam1:
       DWORD dwParam2;
       DWORD dwParam3;
}STRUCTSIZE1;
typedef struct
       DWORD dwSize;
       DWORD dwParam1;
       DWORD dwParam2;
       DWORD dwParam3;
       DWORD dwParam4;
}STRUCTSIZE2;
typedef struct
       DWORD dwSize;
       DWORD dwParam1;
       DWORD dwParam2;
       DWORD dwParam3:
       DWORD dwParam4;
       DWORD dwParam5;
}STRUCTSIZE3;
void FuncGetData (DWORD dwKind, void* lpBuf)
       switch(dwKind)
       }
       case 1:
               //IpBuf IS TREATED AS THE POINTER TO STRUCTSIZE1
               break;
       case 2:
               //IpBuf IS TREATED AS THE POINTER TO STRUCTSIZE2
               break;
       case 3:
               //IpBuf IS TREATED AS THE POINTER TO STRUCTSIZE3
               break;
       }
}
```

FIG. 23



```
290
typedef [public] struct
      [custom (PAT_PARAM_ATTR_ID, "structsize_is()")]DWORD dwSize;
      DWORD dwParam1;
      DWORD dwParam2;
      DWORD dwParam3;
      DWORD dwParam4;
      DWORD dwParam5;
)STRUCTSIZE; \sim 291
interface
test
{
      void FuncGetData
                               292
            [in] DWORD dwKind,
            [in, out] STRUCTSIZE* lpBuf
      );
};
```

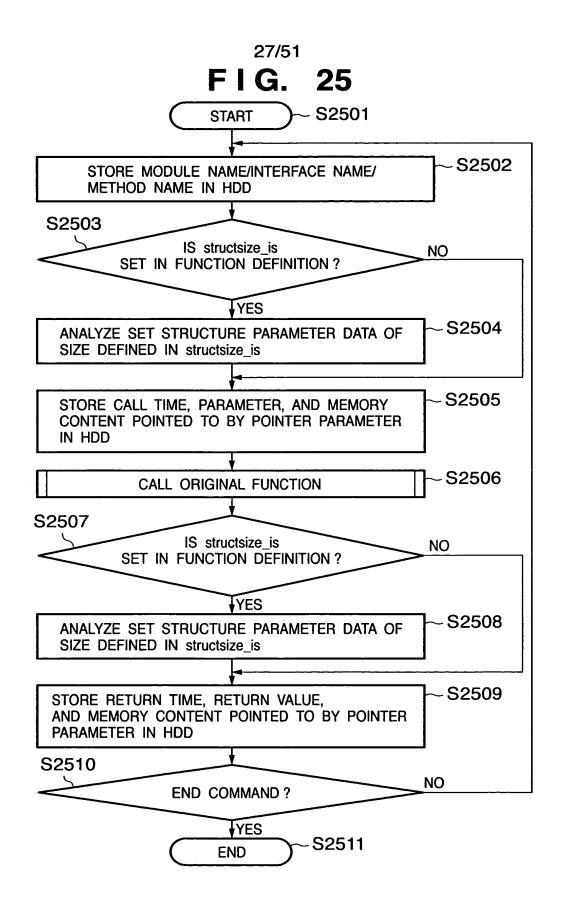
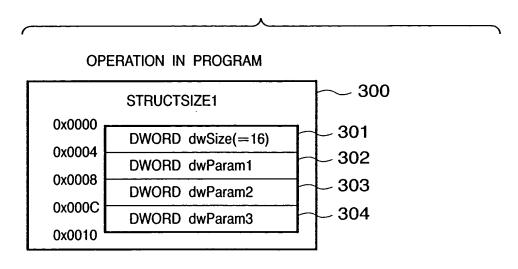
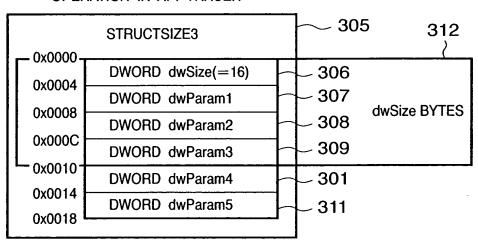


FIG. 26



OPERATION IN API TRACER



MODULE NAME: TestDIIStd. DLL FUNCTION NAME: **FuncGetData**

ARGUMENT(in): DWORD dwKind: 1

STRUCTSIZE* pBuf: 0x503860C DWORD STRUCTSIZE. dwSize: 16 DWORD STRUCTSIZE. dwParam1:0 DWORD STRUCTSIZE. dwParam2:0 DWORD STRUCTSIZE. dwParam3:0

STRUCTSIZE* pBuf: 0x503860C ARGUMENT(out):

DWORD STRUCTSIZE. dwSize: 16 DWORD STRUCTSIZE. dwParam1:1 DWORD STRUCTSIZE. dwParam2:2

DWORD STRUCTSIZE. dwParam3:3

RETURN VALUE: void:

2002/03/25 22:24:12.025 IN TIME: 2002/03/25 22:24:12.035 OUT TIME:

MODULE NAME: TestDIIStd. DLL FUNCTION NAME: FuncGetData DWORD dwKind: 3 ARGUMENT(in):

> STRUCTSIZE* pBuf: 0x503990C DWORD STRUCTSIZE. dwSize: 24

DWORD STRUCTSIZE. dwParam1:0 DWORD STRUCTSIZE. dwParam2:0 DWORD STRUCTSIZE. dwParam3:0 DWORD STRUCTSIZE. dwParam4:0 DWORD STRUCTSIZE. dwParam5:0

STRUCTSIZE* pBuf: 0x503990C ARGUMENT(out):

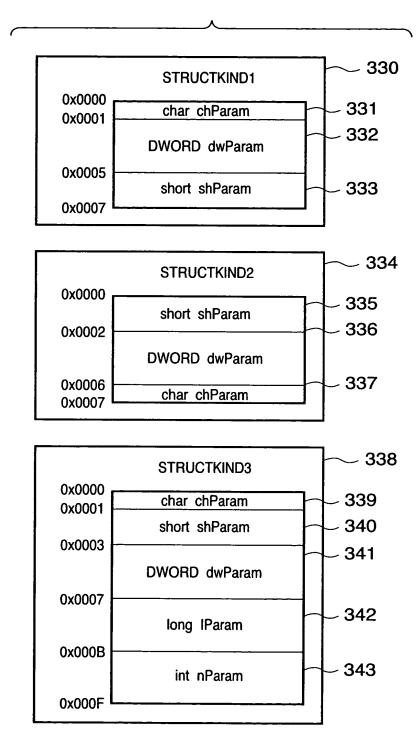
DWORD STRUCTSIZE. dwSize: 24 DWORD STRUCTSIZE, dwParam1: 10 DWORD STRUCTSIZE. dwParam2: 20 DWORD STRUCTSIZE. dwParam3:30 DWORD STRUCTSIZE. dwParam4: 40 DWORD STRUCTSIZE. dwParam5: 50

RETURN VALUE: void:

2002/03/25 22:24:12.046 IN TIME: **OUT TIME:** 2002/03/25 22:24:12.057

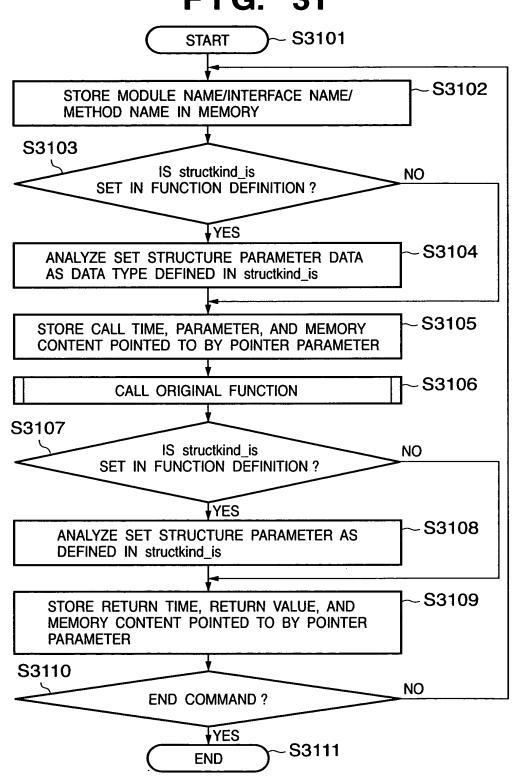
```
typedef struct
       char chParam;
       DWORD dwParam;
       short shParam;
)STRUCTKIND1;
typedef struct
       short shParam;
       DWORD dwParam;
       char chParam;
}STRUCTKIND2;
typedef struct
       char chParam;
       DWORD dwParam;
       short shParam;
       long IParam;
       int nParam;
}STRUCTKIND3;
void FuncGetData (DWORD dwKind, void* lpBuf)
       switch(dwKind)
       case 1:
               //IpBuf IS TREATED AS THE POINTER TO STRUCTKIND1
               break:
       case 2:
               //IpBuf is treated as the pointer to structkind2
               break;
       case 3:
               //lpBuf IS TREATED AS THE POINTER TO STRUCTKIND3
               break;
       }
}
```

FIG. 29



```
#define PAT_PARAM_ATTR_ID 00000000-0000-0000-000000000000
typedef [public] struct
        char chParam;
        DWORD dwParam;
        short shParam;
)STRUCTKIND1;
typedef [public] struct
        short shParam;
        DWORD dwParam;
        char chParam;
}STRUCTKIND2;
typedef [public] struct
        char chParam;
        short shParam;
        DWORD dwParam;
        long IParam;
        int nParam;
)STRUCTKIND3;
interface
test
        void FuncGetData
                [in] DWORD dwKind,
                [in, out, custom(PAT_PARAM_ATTR_ID,
                "structKind_is(dwKind: 1: STRUCTKIND1*, 2: STRUCTKIND2*, 3: STRUCTKIND3*)")]
                void* lpBuf
        );
};
```

33/51 **FIG. 31**



MODULE NAME: TestDII FUNCTION NAME: FuncG

ARGUMENT(in):

TestDIIStd. DLL FuncGetData

DWORD dwKind: 1

STRUCTKIND1* pBuf: 0x503860C char STRUCTKIND1. chParam: 0 DWORD STRUCTKIND1. dwParam: 0

short STRUCTKIND1. shParam: 0

ARGUMENT(out):

STRUCTKIND1* pBuf: 0x503860C char STRUCTKIND1. chParam: 1 DWORD STRUCTKIND1. dwParam: 2 short STRUCTKIND1. shParam: 3

RETURN VALUE:

void:

IN TIME:

2002/03/25 22:24:12.025 2002/03/25 22:24:12.035

MODULE NAME:

TestDIIStd. DLL

FUNCTION NAME:

FuncGetData

ARGUMENT(in):

DWORD dwKind: 3 STRUCTKIND3* pBuf: 0x503990C

char STRUCTKIND3. chParam: 0 short STRUCTKIND3. shParam: 0 DWORD STRUCTKIND3. dwParam: 0

long STRUCTKIND3. IParam: 0 int STRUCTKIND3. nParam: 0

ARGUMENT(out):

STRUCTKIND3* pBuf: 0x503990C char STRUCTKIND3. chParam: 10 short STRUCTKIND3. shParam: 20 DWORD STRUCTKIND3. dwParam: 30

long STRUCTKIND3. IParam: 40 int STRUCTKIND3. nParam: 50

RETURN VALUE:

void:

IN TIME:

2002/03/25 22:24:12.046

OUT TIME:

2002/03/25 22:24:12.057

• • •

FIG. 33

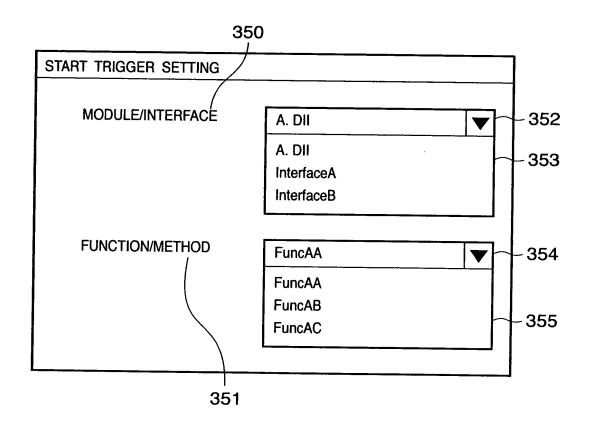


FIG. 34

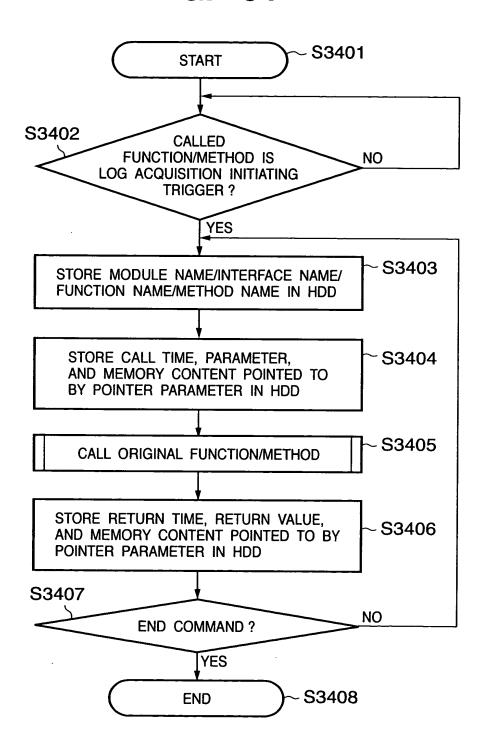


FIG. 35

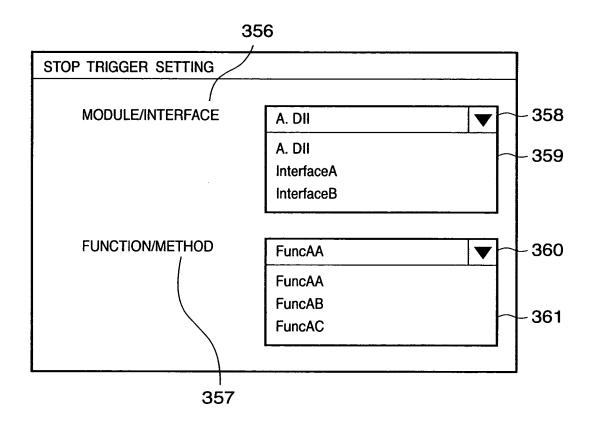


FIG. 36

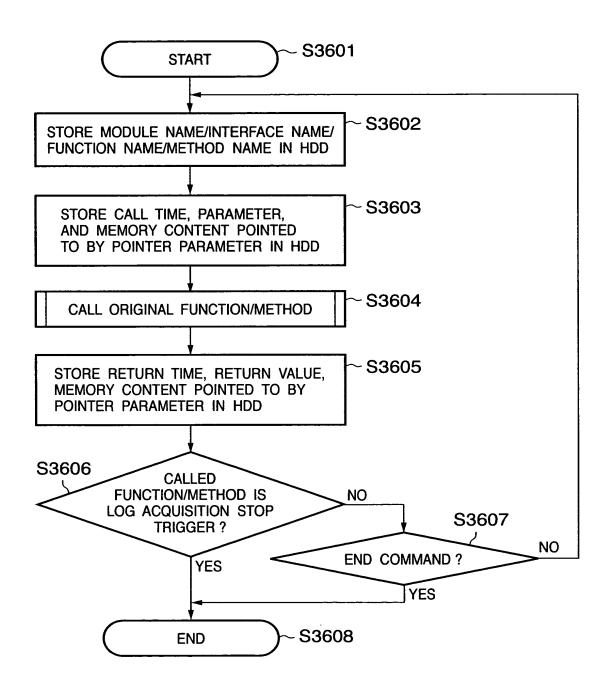


FIG. 37

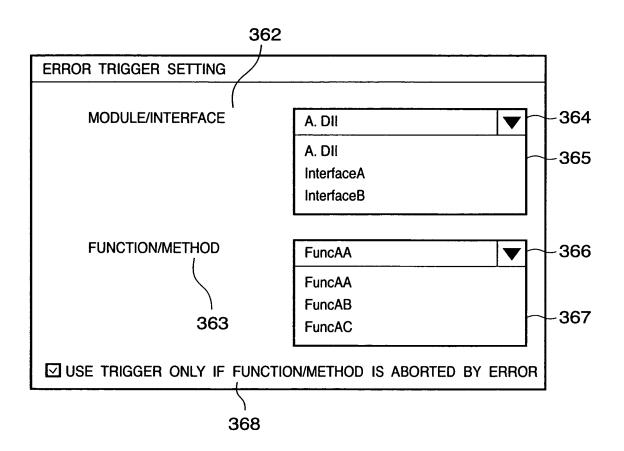


FIG. 38

MODULE NAME:

FUNCTION NAME:

ARGUMENT:

RETURN VALUE:

A. DLL

FuncAA

DWORD dwlD: Err>100

DWORD dwRet: Err==0

MODULE NAME:

FUNCTION NAME:

FUNCTION NAME

ARGUMENT : RETURN VALUE : A. DLL

FuncAB DWORD dwHandle: Err==0

int nRet: Err < = -1

MODULE NAME:

INTERFACE NAME :

METHOD NAME:

ARGUMENT:

B. DLL

InterfaceA MethodAA

DWORD dwID: Err>100

RETURN VALUE: DWORD dwHandle: Err==0

MODULE NAME:

INTERFACE NAME:

METHOD NAME:

B. DLL InterfaceA

MethodAB

ARGUMENT:

 $DWORD\ dwID: Err <= 0$

RETURN VALUE : DWORD dwRet: Err!=0

MODULE NAME:

INTERFACE NAME :

METHOD NAME:

B. DLL

InterfaceB

ARGUMENT : RETURN VALUE : MethodBA DWORD dwID: Err>=0

DWORD dwRet: Err!=0

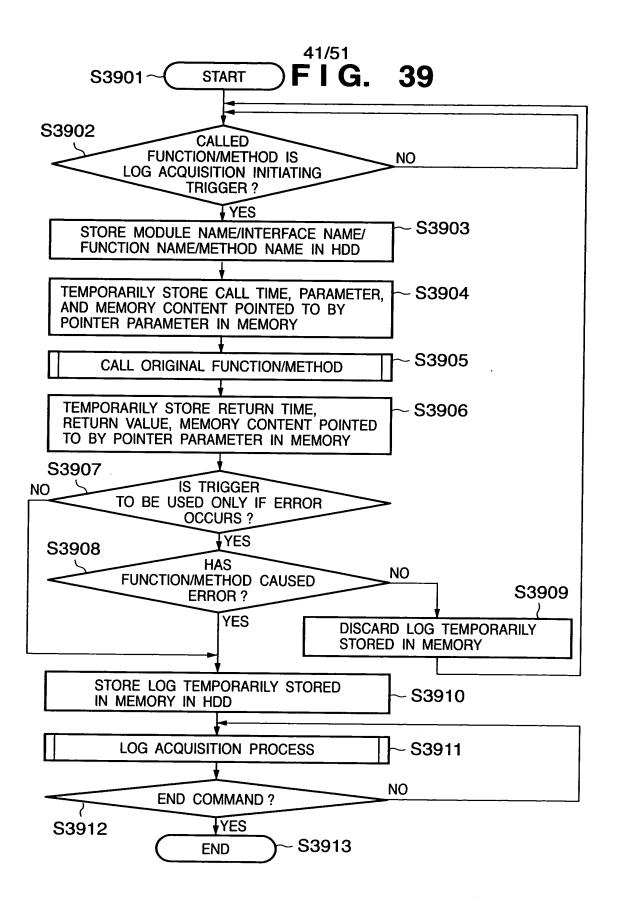
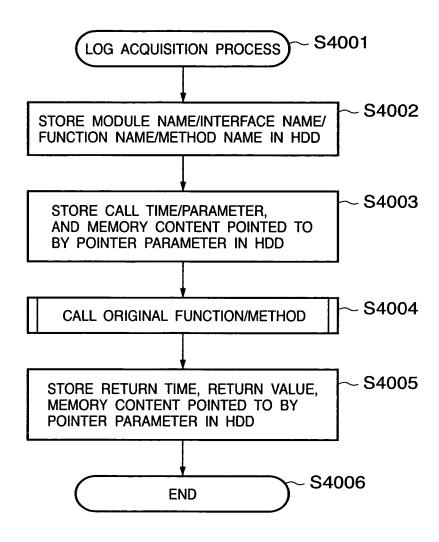


FIG. 40



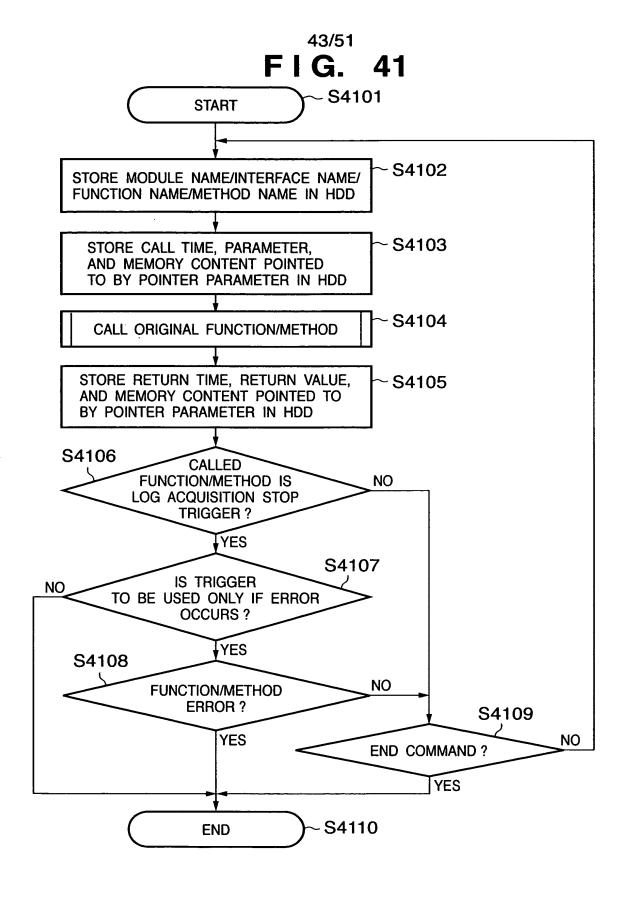


FIG. 42

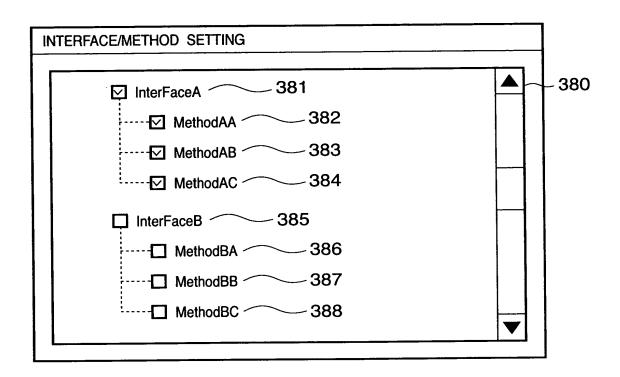


FIG. 43

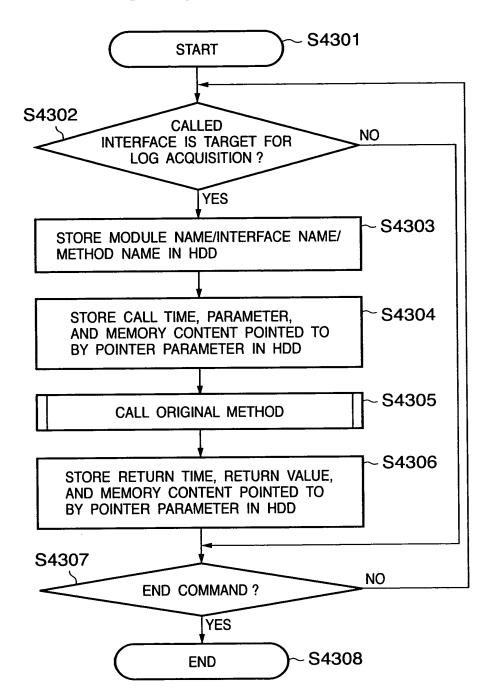


FIG. 44

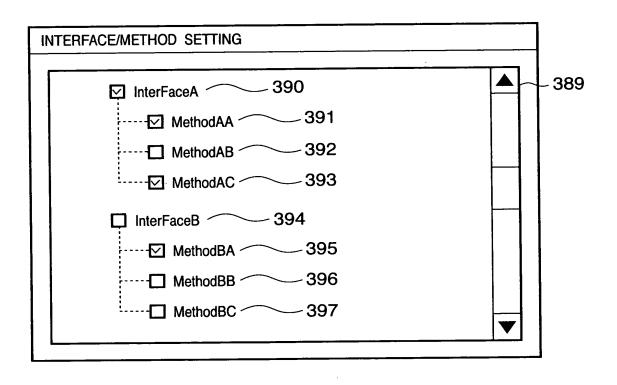
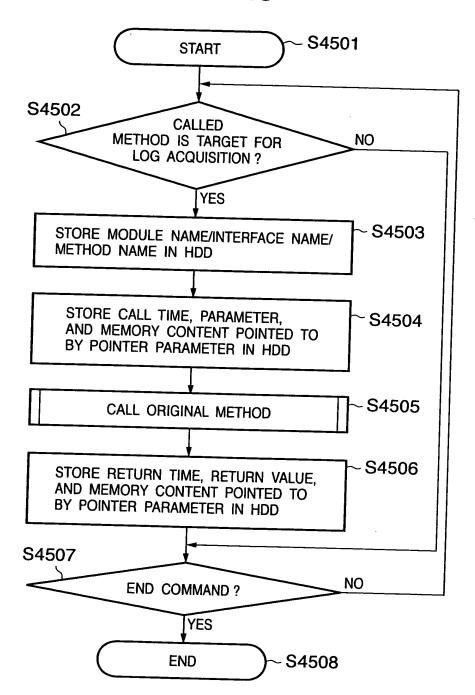
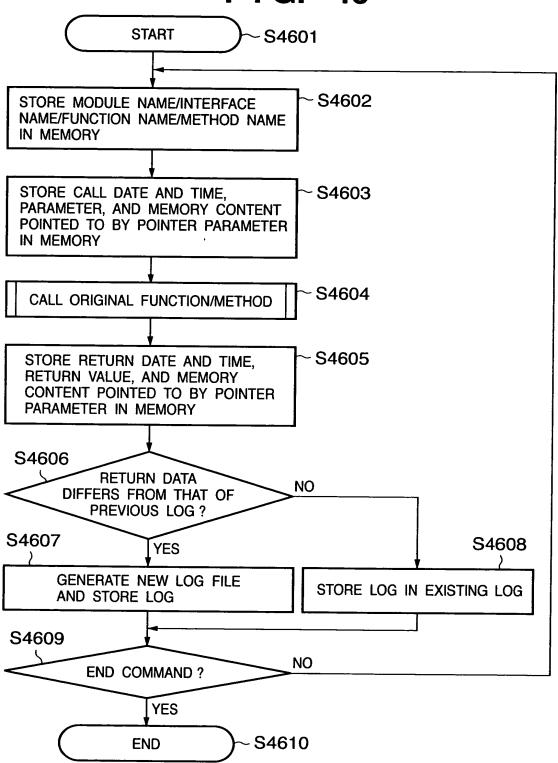


FIG. 45



48/51

FIG. 46



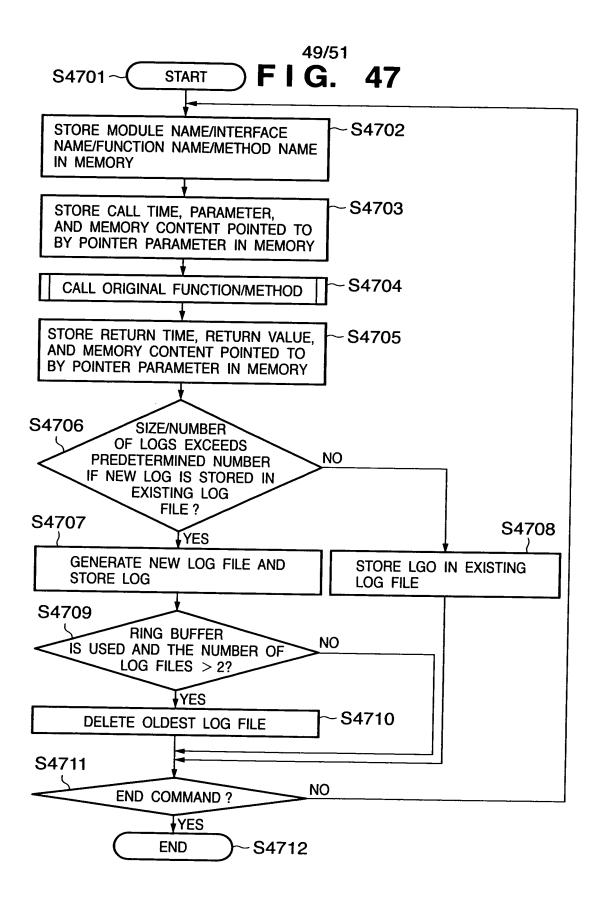


FIG. 48

398 399 MODULE NAME LOG STORAGE MEMORY AREA 1 INTERFACE NAME FUNCTION/METHOD NAME CALL TIME LOG STORAGE MEMORY AREA 2 PARAMETER DATA AT CALL TIME END TIME PARAMETER DATA AT END TIME LOG STORAGE MEMORY AREA 3 RETURN VALUE DATA LOG STORAGE MEMORY AREA 4 LOG STORAGE MEMORY AREA n-1 LOG STORAGE MEMORY AREA n

